REMARKS

The present amendments and remarks are responsive to an Office Action mailed June 17, 2005, where the Examiner has maintained the rejection of claims 1-58. Herein, applicant has made no claim amendments, so no listing of the claims is provided. Reconsideration and allowance of pending claims 1-58 is respectfully requested in view of the following remarks.

A. Response to Provisional Rejection of Claim 1

In the office action, the Examiner rejects claim 1 under the nonstatutory judicially created doctrine of double patenting as being unpatentable over claim 1 of co-pending application number 09/916,900.

Since this is a provisional rejection, the applicant respectfully requests that it be allowed to withhold its response until claims have issued from the 09/916,900 application, and claim 1 in this application has been found to have allowable subject matter.

B. Response to Rejection of Claims Under 35 U.S.C. §103

In the office action, the Examiner rejects claims 1-58 under 35 U.S.C. §103(a) as being unpatentable over U.S. patent 6,442,660 ("Henerlau") in view of U.S. patent 6,023,620 ("Hansson").

1. Claim 1.

The applicant respectfully traverses the rejection of claim 1, and submits that the Examiner has not established a *prima facie* case of obvious as to claim 1. More specifically, Henerlau and Hansson, whether considered singly or in combination, fail to teach all the limitations of claim 1.

In citing to Henerlau, the Examiner finds that Henerlau discloses "receiving, resizing, and executing a plurality of code sections", and specifically cites to figures 2 and 3, as well as three specific passages in Henerlau. The applicant respectfully submits that Henerlau in general, and the cited items in particular, do not disclose any structure that 1) stores system software in a plurality of code sections; 2) receives any new code section; 3) resizes any code section; or 4) arranges a new code section with a current code section to form updated system software.

First, Henerlau does not disclose storing system software in a plurality of code sections. Instead, the code is illustrated in Figs. 2 and 3 as being in a single, continuous file, and at col. 4, lns. 10-20, the code file is referred to as a hex file. There is no indication that the code is stored, used, accessed, or in any way organized according to a plurality code sections.

Second, Henerlau does not receive any updated code sections. As discussed above, Henerlau does not even address an organization of code sections, but more importantly, Henerlau is incapable of receiving any updated code section. For example, Henerlau is directed to embedded devices where the application and code is frequently "permanently stored in ROM", and therefore is not able to be updated. *(Henerlau, col. 3, lns. 5-7)*. Further, as stated in Henerlau, the method of Henerlau creates a single version of an embedded application, which may be executed from either ROM or RAM (*Henerlau, col. 2, lns. 47-50*). This single version of the code is executed every time the Henerlau device is initialized, although it may be executed from RAM or ROM, depending on the amount of RAM available. *(Henerlau, col. 4, lns. 3-10)*. In fact, the ROM version of the code and the RAM representation of the code "need to be nearly identical". *(Henerlau, col. 5, lns. 45-49)*. Accordingly, the device and processes of Henerlau do not teach receiving an updated code, and further, any teachings of Henerlau would require that code differences be "nearly identical".

Third, Henerlau fails to teach any resizing of code. As set out in detail in col. 4, lines 12-37, the process of Henerlau has two representations of the same version of code: one that operates from ROM, and one that operates from RAM. Each of the representations of code has a size that is set during initialization, which does not change. The two code files are compared, and a separate file is created that identifies the differences between the two code versions. Thus, neither of the code files has its size adjusted, and even more particularly, no current code section is resized. The ROM code file has a set size, the RAM code file has a set size, and a difference file logs any differences between the two files.

In the office action, the Examiner remarks that "Henerlau clearly discloses resizing the code section by adjusting the heap pointer", and cites to col. 8, lines 47-55 for support. *(See the office action, pg. 20, lns. 4-10).* However, the applicant submits that the cited passage does not disclose any resizing of the code section, but instead simply moves the existing RAM code file into available RAM memory, and sets the heap pointer according to the existing and static size of the RAM code. For example, the cited section states that "the initialization logic copies the code section into the area of memory just above the data section in RAM". *(Henerlau, col. 8, lns. 50-51).* As described, it is a mere copying process, and does not involve any resizing. Henerlau continues: "The heap pointer is increased by an amount equivalent to the size of the code section." *(Henerlau, col. 8, lns. 51-52).* In this way, the heap pointer is able to identify a free memory area. *(See Fig. 3 (remaining RAM), and col. 4, lns. 62-67):*

> *The total amount of free memory is decreased by the amount of RAM needed for the application. The technique of conditionally inserting the copy of the program at the beginning of the heap memory, and adjusting the position of the heap memory, along with its reference, is an important feature of the invention*).

In this way, the adjustment of the heap pointer is not to accommodate any resizing of the code file, but is simply used to identify the start of a free memory area. Stated differently, the code file has not change size, but has been relocated into the RAM area. Henerlau continues: "the relocation table is read from its well known location in ROM...". This further indicates that size and location values are fixed, and not subject to any "resizing", as set out in claim 1.

Fourth, Henerlau fails to teach the arranging of a new code section with a current code section to form updated system software. Instead, Henerlau has two files capable of executing the same operating code, and simply operates one or the other depending on current memory conditions. Neither version of the code is updated according to any new code section.

Hansson fails to overcome the deficiencies identified in Henerlau. For example, Hansson does not address storing system software in code sections, instead providing for the download of an entire operational version of the operating software into a separate memory. After the new operational software is stored in a memory, the wireless device of Hansson can switch between using the existing operating software or the new operating software. (*See, Hansson, col. 2, lns. 24-40)*. The new software of Hansson is downloaded, received, stored, and executed without suggesting the use of a plurality of code sections, as claimed by the applicant.

Also, since Hansson simply toggles or switches between two available operating systems in separate memories, Hansson also fails to disclose or suggest the arranging of a new code section with current code sections to form an updated system software. Instead, in Hansson, the new software and the current software are entirely predefined at manufacture or prior to download, and therefore do not allow for any dynamic arrangement processes.

For the reasons described above, the applicant respectfully submits that Henerlau and Hansson, either alone or in combination, fail to disclose or suggest all the limitations of claim 1. Accordingly, Henerlau and Hansson can not render obvious claim 1, or its dependent claims 2-27.

2. Claim 28.

The applicant respectfully traverses the rejection of claim 28, and submits that the Examiner has not established a *prima facie* case of obvious as to claim 28. More specifically, Henerlau and Hansson, whether alone or in combination, fail to teach all the limitations of claim 28. For the reasons set out in Section 1(A) above, the applicant submits that neither Henerlau nor Hansson disclose any structure that 1) stores system software in a plurality of code sections; 2) receives any new code section; 3) resizes any current code section; or 4) replaces a current code section with a new code section to form updated system software. Accordingly, Henerlau and Hansson can not render claim 28 obvious.

3. Claim 29.

The applicant respectfully traverses the rejection of claim 29, and submits that the Examiner has not established a *prima facie* case of obvious as to claim 29. More specifically, Henerlau and Hansson, whether alone or in combination, fail to teach all the limitations of claim 29. For the reasons set out in Section 1(A) above, the applicant submits that neither Henerlau nor Hansson disclose any structure that 1) stores system software differentiated into a plurality of code sections; 2) receives any new code section; 3) resizes any current code section; or 4) provides for arrangement of a new code section with a current code section to form updated system software. Accordingly, Henerlau and Hansson can not render obvious claim 29, or its dependent claims 30-57.

4. Claim 58.

The applicant respectfully traverses the rejection of claim 58, and submits that the Examiner has not established a *prima facie* case of obvious as to claim 58. More specifically, Henerlau and Hansson, whether alone or in combination, fail to teach all the limitations of claim 58. For the reasons set out in Section 1(A) above, the applicant submits that neither Henerlau nor Hansson disclose any structure that 1) stores system software differentiated into a plurality of code sections; 2) receives any new code section; 3) resizes any current code section; or 4) provides for arrangement of a new code section with a current code section to form updated system software. Accordingly, Henerlau and Hansson can not render claim 58 obvious.

C. Extension of Time

In accordance with 37 C.F.R. 1.36(a), a credit card payment in the amount of $450.00 is enclosed to cover the Petition for Extension of Time fee set forth under 37 C.F.R. 1.17(a)(1).

D. Conclusion

The applicant believes the pending claims are patentably distinguishable over the cited references. For all the foregoing reasons, an early allowance of claims 1-58 pending in the present application is respectfully requested.

Respectfully submitted,

Dated: Nov 1, 2005

By:
Jonathan T. Velasco, Esq.
Attorney for Applicant
Reg. No.: 42,200

Jonathan T. Velasco, Esq.
Kyocera Wireless Corp.
Attn: Patent Department
P.O. Box 928289
San Diego, California 92192-8289
Tel: (858) 882-3501
Fax: (858) 882-2485